Parte 2 - 12.5 valores

Considere as seguintes definições de classes de uma aplicação que implementa um sistema de gestão de trabalhos práticos de uma unidade curricular. A classe `GestaoTrabalhos` possui a **informação dos alunos** nela registados e dos **grupos de alunos**. Cada grupo conhece as entregas que são feitas na unidade curricular. Cada entrega de uma fase do trabalho prático é avaliada pela equipa docente, mas também por um outro aluno inscrito na unidade curricular, sendo que deverá ser necessário que este aluno não pertença ao grupo que efectua a entrega. Deverá também existir em `GestaoTrabalhos` uma data limite para determinar se uma entrega pode, ou não, ser efectuada. Para poder alterar essa data a classe `GestaoTrabalhos` deverá possuir um método `public void setDataLimite(LocalDate dataLimite)`.

Numa perspectiva mais operacional este sistema funciona como uma unidade curricular dos nossos cursos. Inicialmente os alunos não estão alocados a um grupo e posteriormente existe uma fase de inscrições e só nessa altura é que a informação em que se diz a que grupo pertence cada aluno é que é actualizada.

Considere os seguintes excertos de código:

```
public class Aluno implements Comparable<Aluno>, Serializable {
  private String codAluno;
  private String nomeAluno;
  private Grupo meuGrupo;
  private int notaTeorica;
  private int notaPratica;

  public void regista(Grupo g) {...}
  public int calculaNotaFinal() {...} // calcula a nota final de um aluno
}

public class Grupo implements Comparable<Grupo>, Serializable {
  private String codGrupo;
  private List<Entrega> entregas;

  public Grupo(String codGrupo) {...}

  public void addEntrega(Entrega e) {...}
  public int calculaNotaGrupo() {...}
}

public class Entrega implements Comparable<Entrega>, Serializable {
  private LocalDate data;
  private int nota_docente;
  private Aluno avaliador;
  private int nota_avaliador;
  private String comentarios;

  public int calculaNotaEntrega() {...}  //método que calcula a nota desta entrega
}
```

Assuma, para as perguntas seguintes, que os métodos usuais (get, set, equals, clone, hashcode, ...) estão disponíveis, a menos que sejam solicitados, e responda às questões:

Questão 6

Efectue a declaração das variáveis de instância de `GestaoTrabalhos`, sabendo que esta deverá ter acesso aos alunos e aos grupos e **justifique** a escolha das estruturas de dados que faz. Considere que a estratégia de associação entre a `GestaoTrabalhos` e os alunos e grupos é de composição e entre Aluno e Grupo é de agregação (partilha o apontador) conforme está patente no código da classe Aluno.

Codifique também o método `public void adicionaAluno(Aluno a) throws ...` , da classe `GestaoTrabalhos`, que adiciona um aluno, verificando que o mesmo não exista (não necessita de fazer o código da classe de excepção).

☐0 ☐.2 ☐.4 ☐.5 ☐.6 ☐.8 ☐1 *Reservado aos docentes*

Questão 7

Codifique o método `public Entrega melhorEntrega()`, que determina a `Entrega` com melhor nota de todas as existentes no sistema. Caso exista mais do que uma entrega com a mesma nota deverá ser devolvida aquela cujo código de grupo seja alfabeticamente menor.

☐0 ☐.2 ☐.4 ☐.5 ☐.6 ☐.8 ☐1 *Reservado aos docentes*

Questão 8

Codifique o método `public void adicionaEntrega(String codGrupo, Entrega e) throws ...`, da classe `GestaoTrabalhos`, que deverá inserir uma entrega nas entregas de um grupo. Esta operação só deverá ser possível caso o aluno que aparece como avaliador na entrega não pertença ao grupo indicado como parâmetro e a data limite que está definida não tenha sido ultrapassada.

☐0 ☐.2 ☐.4 ☐.5 ☐.6 ☐.8 ☐1 *Reservado aos docentes*

Questão 9

Para a classe `GestaoTrabalhos` codifique o seu construtor, `public GestaoTrabalhos(Collection<Aluno> alunos, Map<String,String> alunosGrupo, LocalDate dataLimite)`, que recebe uma série de instâncias de `Aluno` e um `Map<String,String>` que contém uma relação `Numero Aluno -> Código do seu grupo` e cria uma instância válida de `GestaoTrabalhos`, com todos os alunos e grupos, respeitando a associação entre alunos e grupos expressa no Map. Note que os grupos terão de ser criados neste construtor.

☐0 ☐.2 ☐.4 ☐.5 ☐.6 ☐.8 ☐1 *Reservado aos docentes*

Questão 10

A classe `GestaoTrabalhos` não possibilita o registo de entregas quando a data da mesma é posterior à data limite definida em cada altura pelo sistema. Pretende-se agora ter uma nova classe `GestaoRelaxadaTrabalhos`, que para as entregas tardias as coloque numa estrutura de dados indexadas por código do grupo, de acordo com o excerto de código abaixo.

```
public class GestaoRelaxadaTrabalhos extends GestaoTrabalhos {
   private Map<String,List<Entrega>> entregasAtrasadas;

   ...
}
```

Para esta nova classe codifique o método `public void adicionaEntrega(String codGrupo, Entrega e)`, por forma a permitir também registar as entregas tardias. Como medida de salvaguarda este método deverá também criar um ficheiro de objectos, para cada Entrega, em que o nome do ficheiro é a concatenação do código do grupo com a data da entrega.

☐0 ☐.2 ☐.4 ☐.5 ☐.6 ☐.8 ☐1 *Reservado aos docentes*